Course Overview
Machine Learning Introduction

BA810: Supervised Machine Learning

Nachiketa Sahoo

What is Machine Learning?

- Machine learning is the "field of study that gives computers the ability to **learn without being explicitly programmed**" (Arthur Samuel 1959).
- "A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P* if its performance at tasks in *T*, as measured by *P*, **improves with experience *E*.**" (Tom Mitchell)
  - T: detecting spam
  - P: percentage of spam messages correctly identified
  - E: labelled spam/non-spam email messages

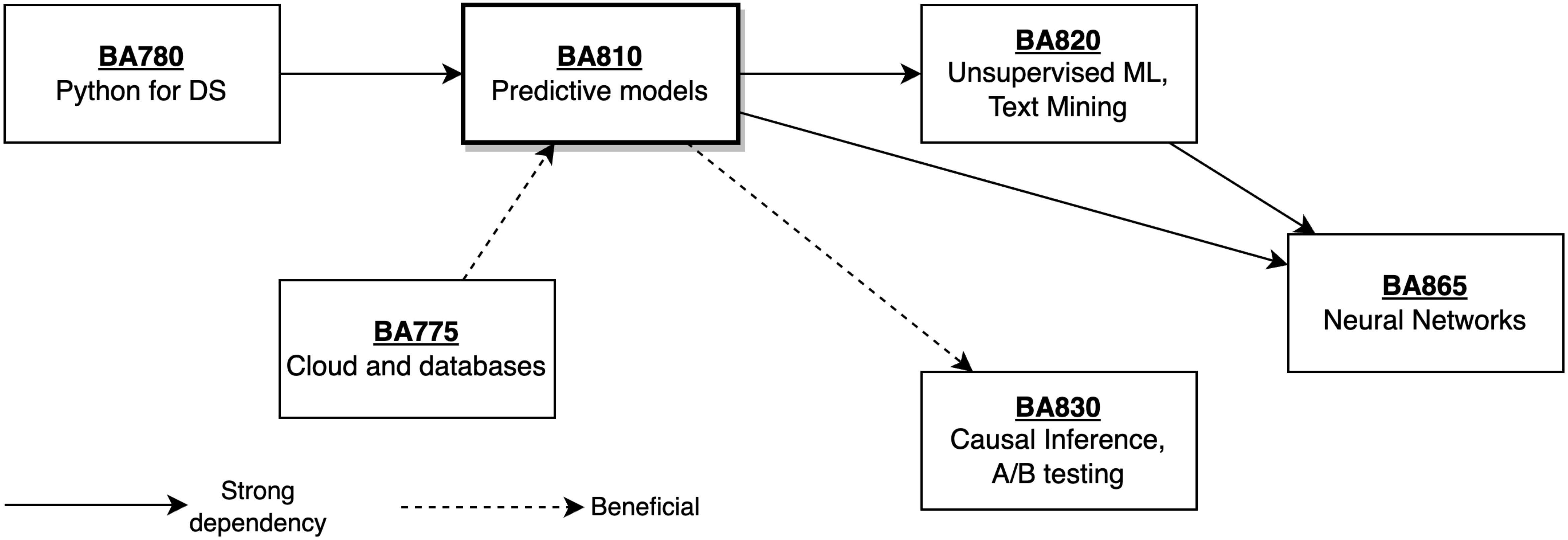
Examples of Machine Learning Problems

- Loan default prediction
- Customer churn prediction
- Fraud detection
- Recommender systems (if you like X you'll enjoy Y)
- Medicine (e.g., radiology, detecting disease from scans)
- Self-driving cars (given a sequence of video frames what would the driver do next (brake, accelerate, swerve, etc.))
- Language translation (e.g., Google translate)
- Weather prediction
- **Others from your experience?**

Today’s Class

1. Course overview
2. Types of machine learning?
3. How do we measure prediction accuracy?

Link to Some Other MSBA Courses



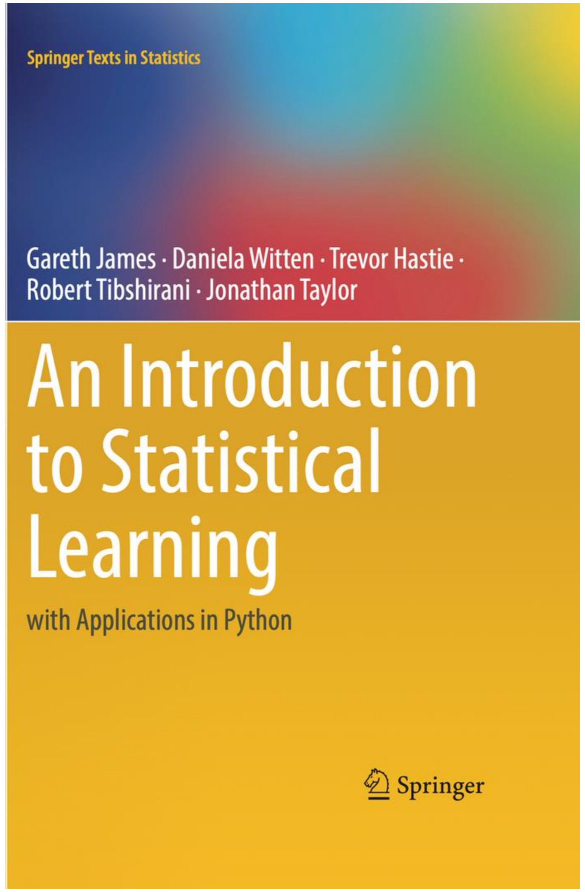
Class Structure

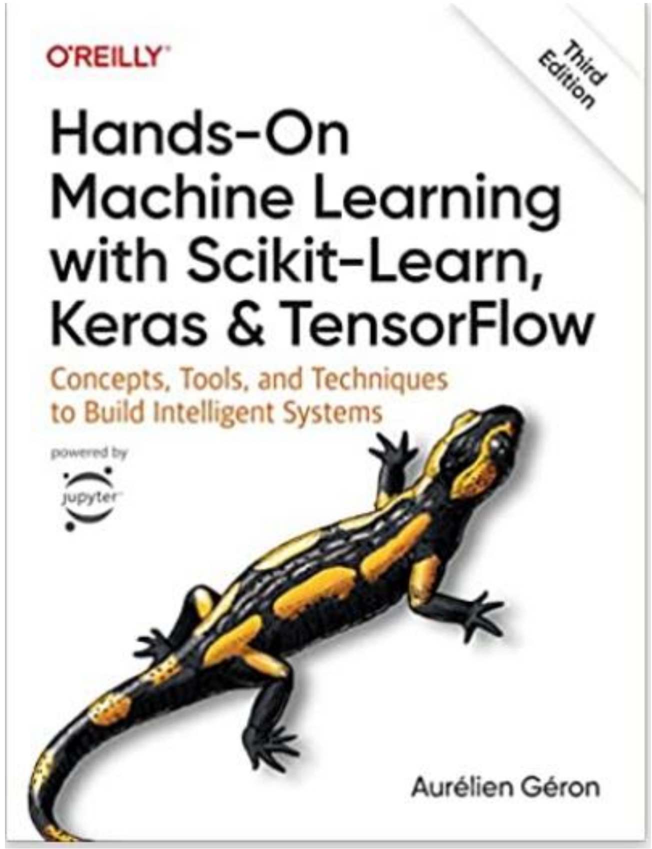
- Applied introduction to ML
  - Lots of work with Python
  - Real world and realistic datasets
- Each class
  - Quick recap
  - Day’s topic discussion (read/do the assigned material before class)
  - (10min break approximately in the middle)
  - Coding and walking through code together
  - Do the code exercises in class

Course Map (Topics)

1. Introduction to Machine Learning
2. General predictive models
    - regression and classification
3. Model selection
4. End-to-end Machine Learning process
5. Specific predictive models
    - Support Vector Machines, Decision Tree, Ensembles
6. Managing imbalanced data in practice

Books


Springer Texts in Statistics
Gareth James · Daniela Witten · Trevor Hastie ·
Robert Tibshirani · Jonathan Taylor
An Introduction to Statistical Learning
with Applications in Python
Springer


O'REILLY
Third Edition
Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow
Concepts, Tools, and Techniques to Build Intelligent Systems
powered by
jupyter
Aurélien Géron

Deliverables

Attendance:	5%
Class participation:	10%
Individual assignments: 2 × 10% =	20%
Datacamp assignments:	15%
Team project: 5% (proposal) + 15% (final) =	20%
Final Exam:	30%
Total:	100%

Teams for Project

- Four students per team
- Place the team number next to all members' names here
  - Need BU Google account to access
  - Two sheets for two sections (A: morning, B: afternoon)
- After today (10/23), I'll assign the unassigned to a team
- Team Learning tool for individual feedback and assessment
- Project/ProjectInstructions.pdf has detailed guidelines
  - Submit only one copy per team (proposal and final slides/notebooks)

Teaching Assistants

Howard Chang

Peiqi Chen

Weiming (Kevin) Wang

Office hours (TAs or myself) on each weekday; see syllabus for details.
Use our slack channel to ask questions, share thoughts/resources.

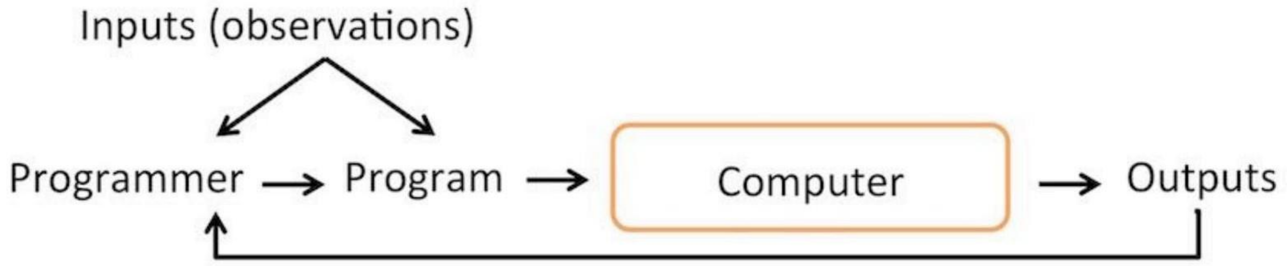
Academic Integrity

- Do not cheat! If you are unsure if certain things are allowed, ask.
- You are allowed to consult each other, and Generative AI, to *learn* while doing the homework and project, but you must:
  - Disclose who you discussed with
  - The prompts you used for GenAI tool (ChatGPT/BARD/Github Copilot)
  - **Write your own code** — can't copy paste code from elsewhere
  - Be able to defend your answers (why done in certain way)
- Applies to anything you submit — must be created/written by you
- You are ultimately individually responsible for what you submit

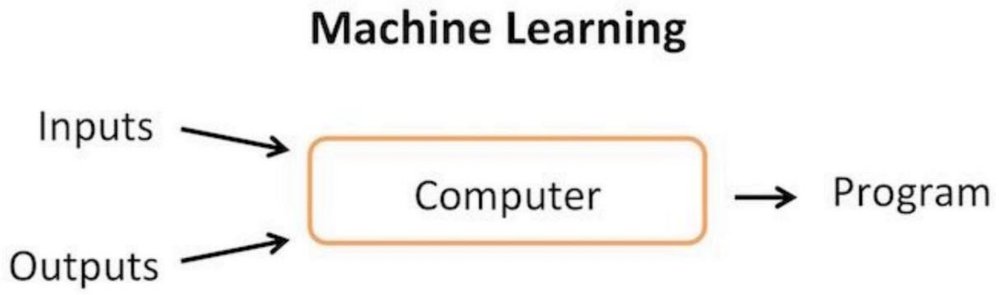
Course Feedback

- What is working well and what can be improved?
  - Talk to me directly
  - Or share anonymously anytime using [this form]

Machine Learning vs Traditional Programming



Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed
– Arthur Samuel (1959)



Different Types of Machine Learning

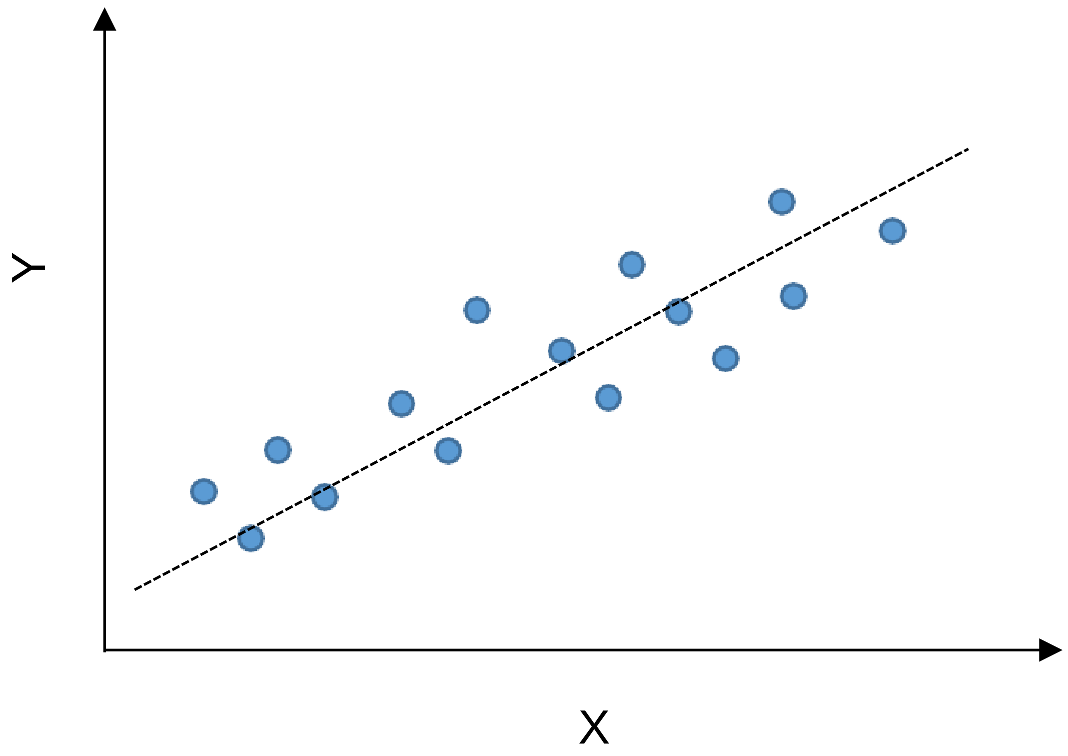
1. Supervised versus unsupervised learning
2. Regression versus classification
3. Prediction versus inference

Supervised Learning

- We are given labelled data with an outcome variable
- $Y = (Y_1, Y_2, \ldots, Y_n)$
  - E.g., sales volume or a label (spam, non-spam)
  - Here, n is the number of observations in our dataset
- For each Y_i , we also have predictors X_i (aka regressors, covariates)
  - E.g., $X_i = (\mathrm{X1}_i, \mathrm{X2}_i, \ldots)$, where X1 is ad spend on TV and X2 is ad spend online
  - Or X could be the words included in an email message
- The goal is to predict Y given X for new unlabeled data

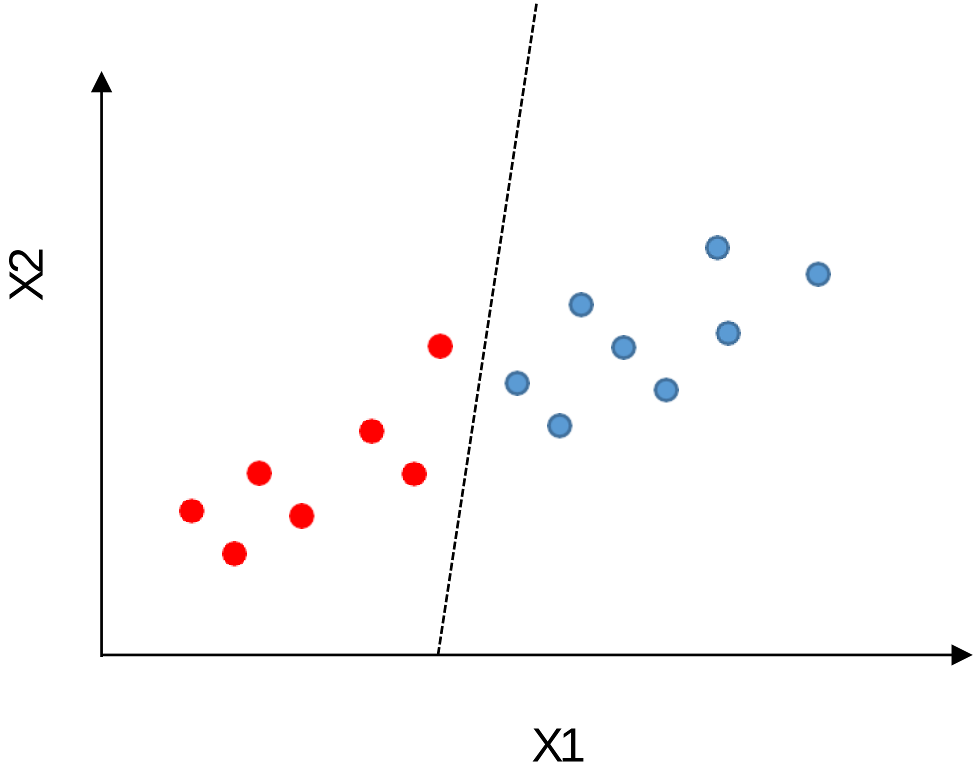
Supervised Learning -- Regression

- Predicting a number



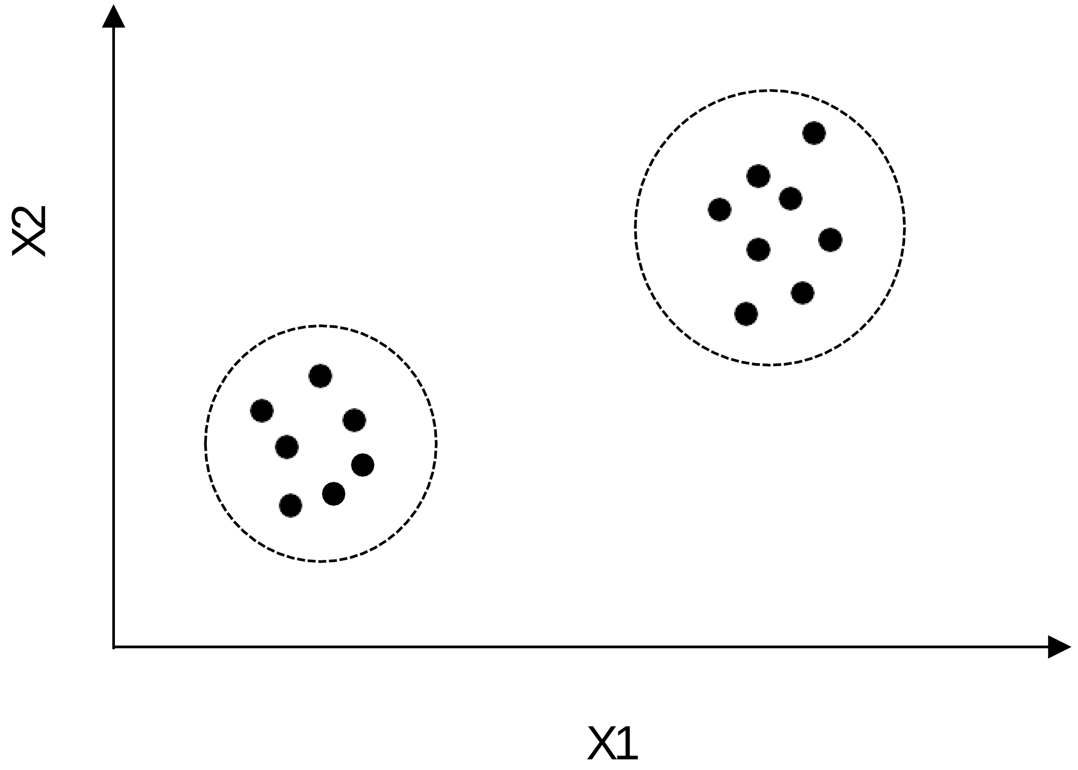
Supervised Learning -- Classification

- Predicting a label



Unsupervised Learning -- Clustering

- There is no outcome Y in our data
- What can we learn in this case?



Objectives in Supervised Learning
Prediction vs Inference

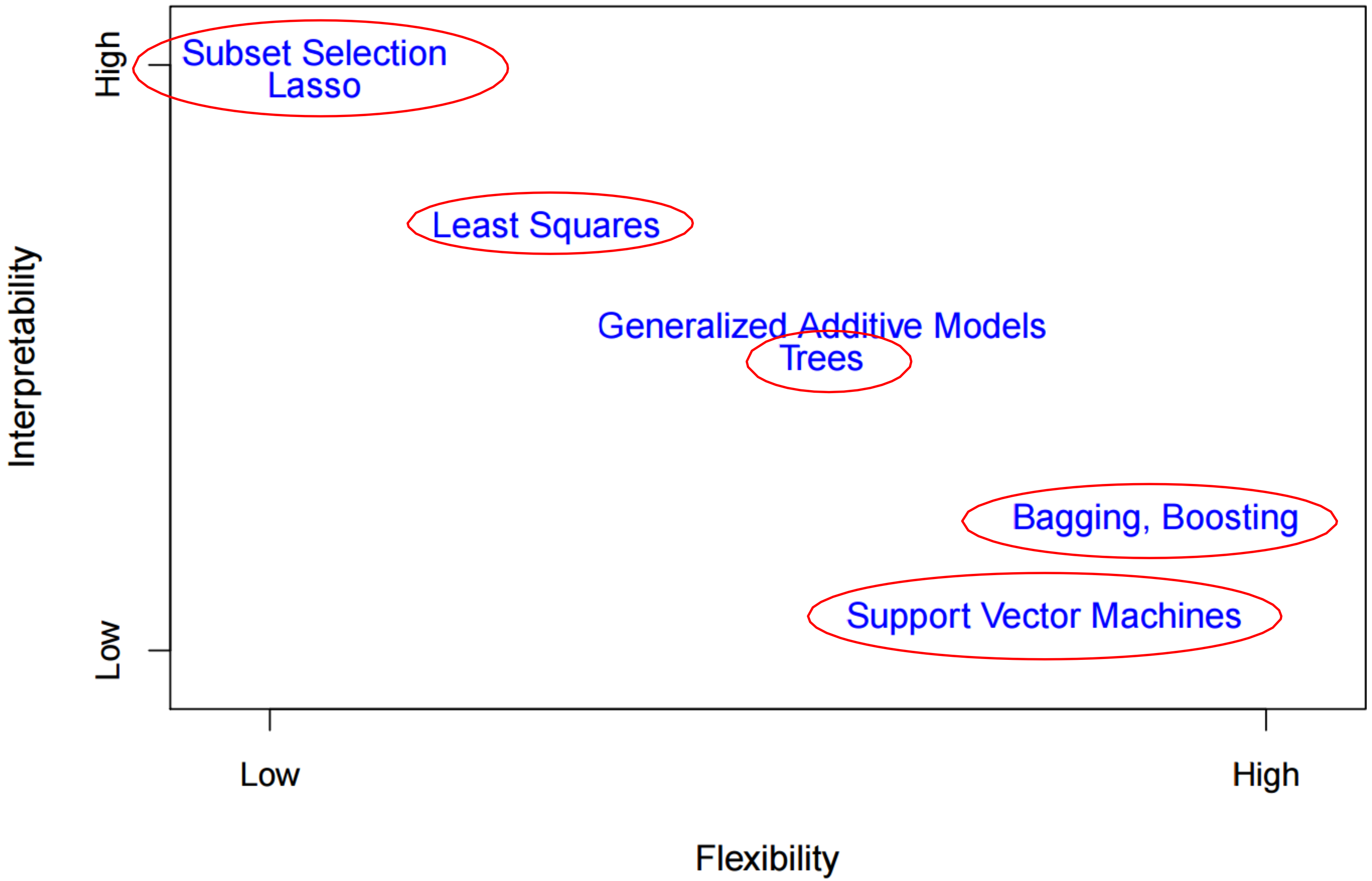
- **Prediction:** Given new data point X, predict a response Y
- $\hat{Y} = \hat{f}(X)$ where $\hat{f}$ is our estimate of f and $\hat{Y}$ is our prediction of Y
    - We don't primarily care what $\hat{f}$ looks like — treat it as a black box
    - The goal is **accurate** prediction of Y given some observed X
- From such $\hat{f}$ we can't say what'd happen if X is manipulated
    - That is causal inference, requires randomized trial (or approximation of it)

Objectives in Supervised Learning
Prediction vs Inference

- **Inference:** Again, we start by estimating $\hat{Y} = \hat{f}(X)$
  - But now we care about the kind of relation between Y and Xs
  - $\hat{f}$ is not a black box any more
- Example: what are the key determinants of customer churn?
  - Reducing these factors to reduce churn
- The goal is **causal inference**
  - To estimate what will happen to Y if we *manipulate* an X
  - Generally, can't draw causal conclusion from predictive models

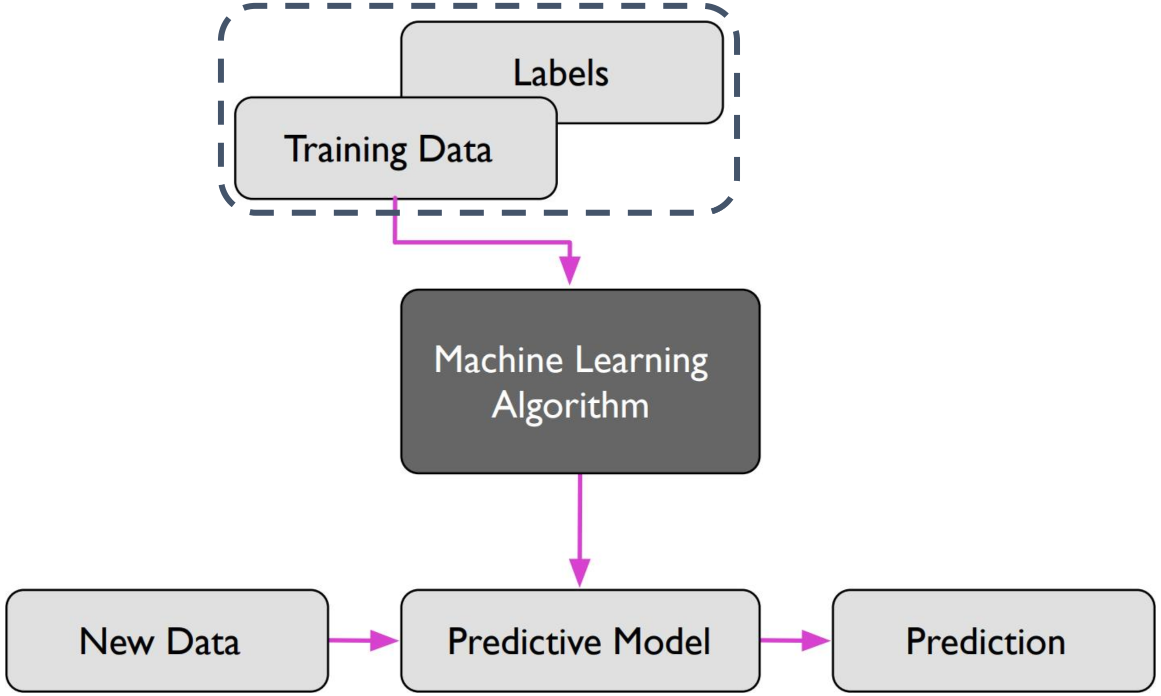
Interpretability or Flexibility

Of predictive models



Source: Tibshirani et al.

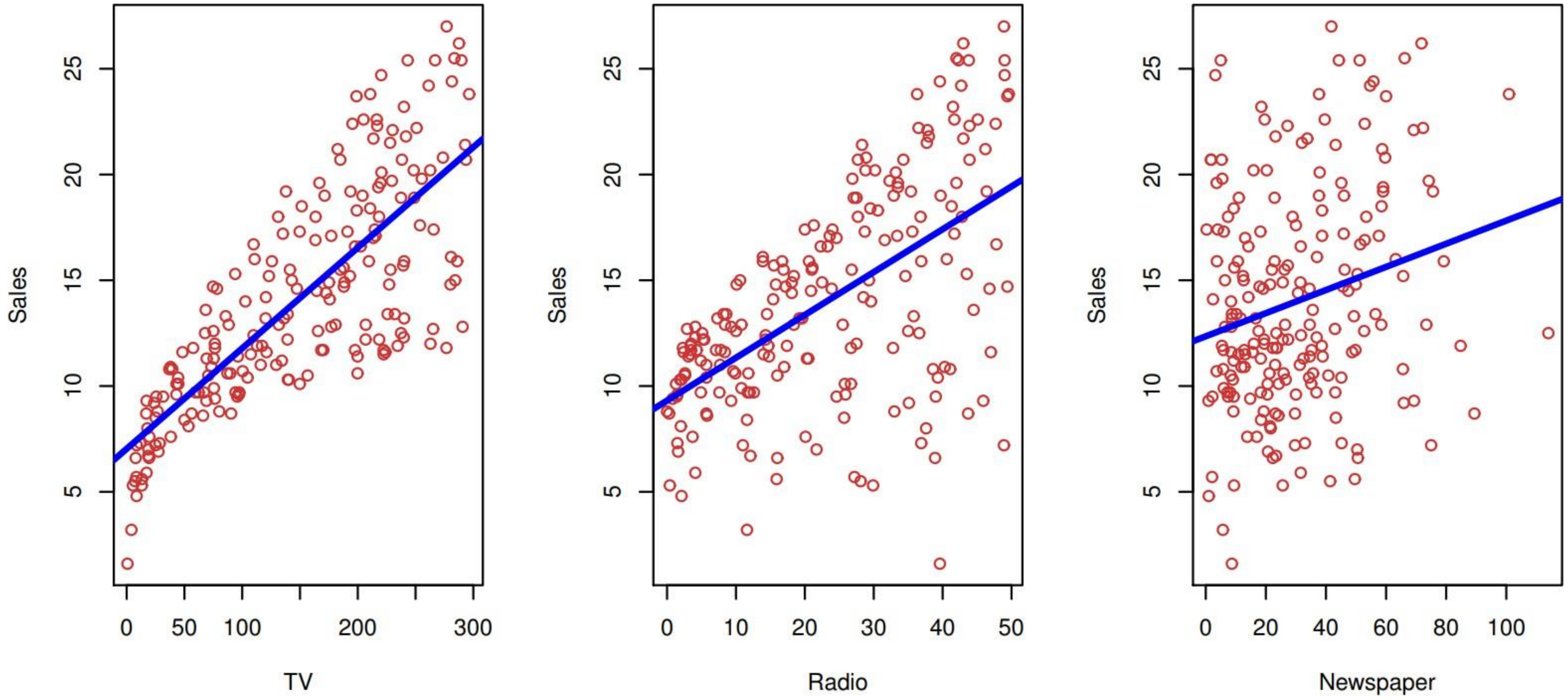
Supervised Learning Workflow



- Define the problem
- Collect labelled training data
  - And clean them
- Choose an ML algorithm, and fit a model to the data
- Evaluate the model according to your chosen metric
- Use the model to predict on new data

Linear Regression

- Simple approach to supervised learning
- Assumes linear relationship between predictors and outcome



Linear Regression

- A simple linear regression model $Y = f(X)$
$$Y = \beta_0 + \beta_1 X_1 + e$$
- β_0: intercept, β_1: slope
- e: unobserved randomness we cannot model ("error")
- Our goal is to estimate the parameters of this model, β_0 and β_1
  - How do we do this?

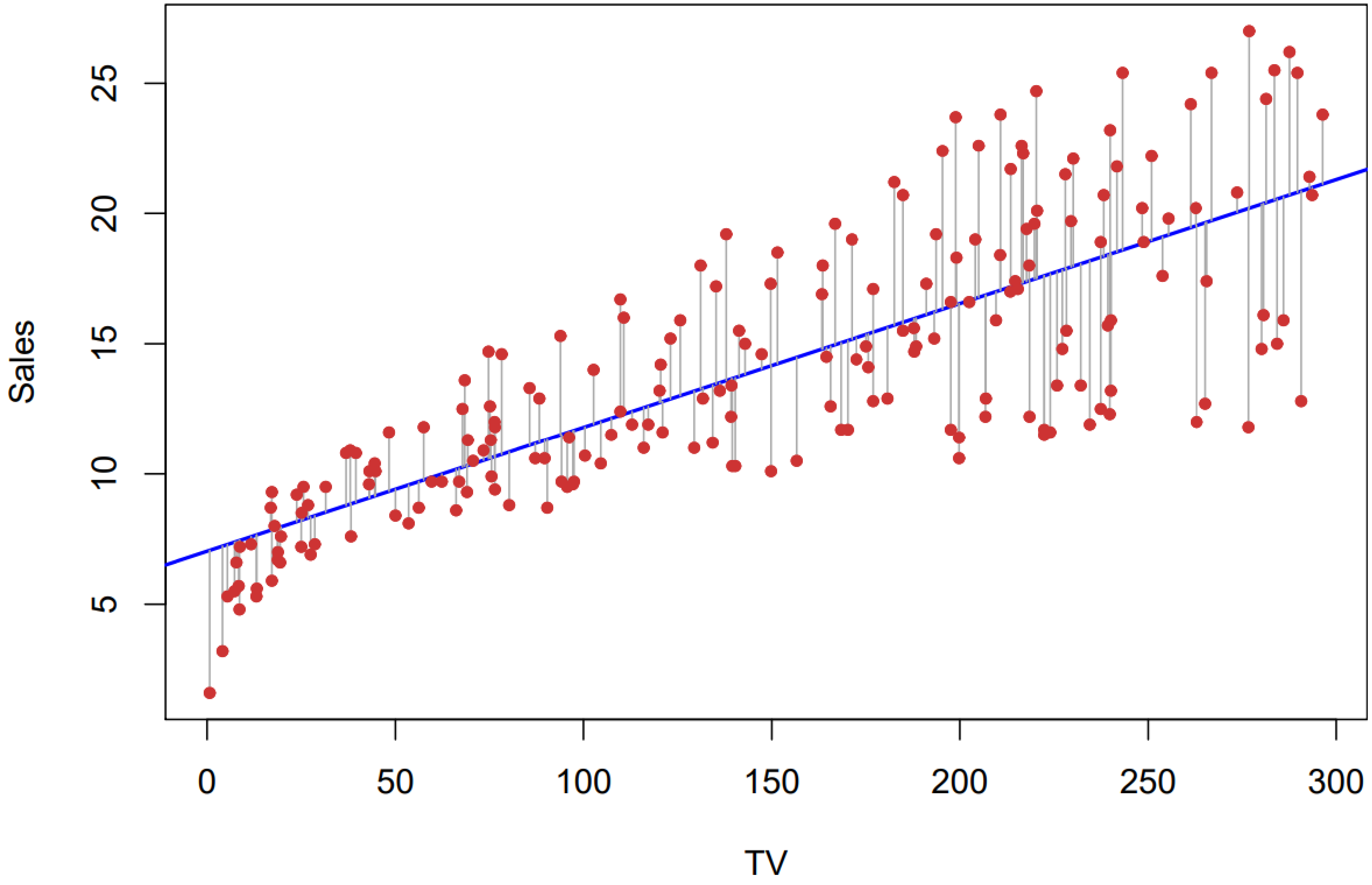
Linear Regression

- A simple linear regression model $Y = f(X)$

$$Y = \beta_0 + \beta_1 X_1 + e$$

- Suppose we have some parameters $\hat{\beta}_0$ and $\hat{\beta}_1$ is the slope
  - Notice the hats — these are parameter estimates from data
- Then we can predict as $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1$
- Measure the error in prediction as $\mathrm{res_i} = Y_i - \hat{Y}_i$ (i'th residual)
  - $\hat{\beta}_0$ and $\hat{\beta}_1$ are chosen to minimize the sum of squares of these residuals

Residuals



Measuring Prediction Performance

- One measure of performance is the average squared residual

$$\text{Mean Squared Error } (MSE) = (res_1^2 + res_2^2 + \cdots + res_n^2)/n$$

  - Why squared?
- Using every observation in your training dataset compute

$$MSE_{train} = \frac{1}{n}\sum_{i}^{n} res_i^2 = \frac{1}{n}\sum_{i}^{n}\left(y_i - \hat{f}(x_i)\right)^2$$

- Can we use this as a measure of prediction performance of the regression model in future data?
  - It'll be too optimistic (error too low) since the model has seen the data

The Train-Test Paradigm

Estimating ***Generalization Error***

- Keep a test dataset that was not used for training

Training set	1						
	2						
	3						
	4						
Test set	5						
	6						

- Compute MSE on test data
  - For every observation in *test data* compute $\mathrm{res}_\mathrm{i}^2 = \left(Y_i - \hat{Y}_i\right)^2$
  - Then compute MSE_{test} by averaging these test residual squares
- Advantages? Drawbacks?

Assessing Prediction Performance

- Mean squared error easier to interpret if taken a square root (why?)
  - RMSE: root mean square error
- How good is an RMSE?
  - For a baseline, fit the simplest model possible: a linear regression just with an intercept term, no x variable — called a *null model*
  - What is the prediction $\hat{y}$ of this model for any observation?
    - Hint: same for all observations
  - What is the RMSE of this null model?

Overview of ML Model Development Process

1. Load and explore data, training-test split
2. Clean and consider transformations
3. Create processes and pipelines
4. Evaluate various predictive models
5. Finetune the most promising model
6. Estimate error on unused test-data

Use only training data
Do not consult the test data — it'll invalidate the test data for estimating error on future data

Use test data
Only to estimate error, not to tune/select model to minimize

- Let's see a first example using scikit-learn
    - Links at "Slides/List of lab notebooks.gdoc"

Summary

- Supervised machine learning
  - Learning how to predict uncertain outcomes from other observed variables (features)
- Type of ML exercises
  - Supervised
    - Classification vs regression
    - For **prediction (BA810)** vs for causal inference (BA830)
    - Measuring **generalization error** — through train-test splitting
  - Unsupervised (BA820)

- First example of ML work using scikit-learn library in Python
  - Review and do the exercises
- Declare teams today
- Regression models next class